


Day 9, Today's Topic

# Const and Let

DAILY  
JS



```
/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */
```

## # Day 9: const and let

Today's topic is rather simple than everything we have discussed so far, and we won't be needing much time on trying to understand it.

Even though it's pretty simple, it still required to be discussed on a separate day in our DailyJS initiative because of its importance. After `const` and `let` were introduced, people rarely use the `var` keyword now, so you can understand how important these two are.

So coming on to the main topic, `const` and `let` are the ways of declaring the variables.

You might be aware of the `var` keyword. Earlier (and it's still acceptable but not recommended) people used to declare variables using `var` keyword.

```
/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```

```
/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */
```

## ## Syntax

Syntax is pretty simple, just like you declare variables in other languages, or even JavaScript with `var`, you can declare variables using `const` and `let`.

Some very important things about them which we should always keep in mind is that

1. They both are `block scoped`,
2. `let` can be reassigned
3. `const` cannot be reassigned

### **const**

```
```js
const <variable_name> = <value>
```
```

### **let**

```
```js
let <variable_name> = <value>
```
```

```
/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```

```

/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */

/**
 * const let example
 */

const name = "Madhav";
let profession = "Student";
let age = 21;

console.log ("Name: ", name);
console.log ("Profession: ", profession);
console.log ("Age: ", age);

// Let's say 1 year passed and his profession changed
// We can reassign
// name = "Madhav new"; → This will give an error
profession = "Software Engineer";
age = 22;

console.log ("/* ===== After a year ===== */");
console.log ("Name: ", name);
console.log ("Profession: ", profession);
console.log ("Age: ", age);

/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */

```

```
/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */
```

### ## What's the problem with `var` though?

To be honest, there were no major problems with `var`, but there were some possibilities of "messing things up".

I know it might sound funny, but it's true.  
Let's consider this example,

```
```js
var name = "Madhav"
console.log (name); // Madhav
var age;
console.log (age); // Undefined. No errors
```
```


Doing anything with uninitialized variables should give an error,  
but it won't

Let's see another example, `var` is not block scoped, before `const` and `let`, variables were either having global scope or function scope, not block scope.

```
```js
var animal1 = "dog";
if (animal1 === "dog") {
  var animal1 = "cat";
}
console.log (animal1); // Should give "dog", but prints "cat"
```
```

```
/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```





```
/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */
```

**\*\*Let's solve both the issues addressed above\*\***

### **## Issue 1 - solved using const**

```
```js
var name = "Madhav"
console.log (name); // Madhav
var age;
console.log (age); // Undefined. No errors

const name1 = "Madhav";
console.log (name1); // Madhav
const age1;
console.log (age1); // Error!
```
```

```
/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```

```

/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */

## Issue 2 - Solved using block scoped let and const

```js
/**
 * var is not block scoped
 * let and const are block scoped
 */


// #1: var
var animal1 = "dog";
if (animal1 === "dog") {
  var animal1 = "cat";
}
console.log (animal1); // cat "Ohh no!"

// #2: let
let animal2 = "dog";
if (animal2 === "dog") {
  let animal2 = "cat";
}
console.log (animal2); // dog "Perfect :)"

// #3 const
const animal3 = "dog";
if (animal3 === "dog") {
  const animal3 = "cat";
}
console.log (animal3); // dog "Perfect :)";
```

/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */

```



```
/* ===== */
/* ===== Daily JS - Day 9 ===== */
/* ===== */
```

## ## Properties of `const`

Before ending today's topic, it's important to put forward some important properties of `const` so that you don't mess things up while using it in future.

`const` can be seen as somewhat similar to constant in other languages.

1. `const` is **\*\*read-only\*\***
2. It can't be redeclared
3. It can't be reassigned
4. Even so, it is mutable,  
Object and array values can be modified.

That's it. See you tomorrow :)

```
/* ===== madhavbahl.tech/dailyjs/day9 ===== */
/* ===== Join Discord: madhavbahl.tech/discord-c2e ===== */
```



# Thank you!

**Feel free to reach out...**

Email: [theleanprogrammer@gmail.com](mailto:theleanprogrammer@gmail.com)

Web: [madhavbahl.tech/](http://madhavbahl.tech/)

Github: [github.com/MadhavBahlMD](https://github.com/MadhavBahlMD)

LinkedIn: [linkedin.com/in/madhavbahl/](https://linkedin.com/in/madhavbahl/)

Insta: [instagram.com/theleanprogrammer/](https://instagram.com/theleanprogrammer/)

**DAILY**  
**JS**